

# 一种新颖的差分混合蛙跳算法<sup>①</sup>

王 娜, 高学军

(广东工业大学 应用数学学院, 广州 510006)

**摘 要:** 在使用智能优化算法处理函数优化问题时, 保持种群的多样性及加快种群的收敛速度可以提升一个算法的性能. 针对混合蛙跳算法在寻优过程中易陷入局部最优和早熟收敛的缺点, 本文提出了一种新颖的差分混合蛙跳算法. 该算法借鉴差分进化中的变异交叉思想, 在前期利用子群中其他个体的有用信息来更新最差个体, 增加局部扰动性, 以提高种群的多样性; 在后期为加快收敛速度使用最好个体的信息进行变异交叉操作. 同时本文使用归档集进一步保留种群的多样性. 仿真测试结果表明: 该算法在求解优化问题时较基本蛙跳算法和平均值蛙跳算法具有更好的寻优性能.

**关键词:** 智能优化; 混合蛙跳算法; 差分进化; 归档集

## Novel Differential Shuffled Frog Leaping Algorithm

WANG Na, GAO Xue-Jun

(Department of Applied mathematics, Guangdong University of Technology, Guangzhou 510520, China)

**Abstract:** When using optimization algorithms to solve optimization problems, keeping the diversity of population and accelerating the convergence rate of the population can improve the performance of an algorithm. To overcome the main drawbacks of the shuffled frog leaping algorithm which may be easy to get stuck and premature convergence in a local optimal solution, this paper proposes a novel differential shuffled frog leaping algorithm. The algorithm is based on the idea of mutation crossover in differential evolution. In the earlier, it uses beneficial information of the other individuals in sub-group to update the worst individual, which increases the local disturbance and the diversity of population; in the later, the algorithm uses the best individual information to conduct the mutation and cross operation for speeding up the convergence rate of the population. Moreover, this paper uses the archive to keep the diversity of population. The experimental results show that the proposed algorithm is superior to the basic frog leaping algorithm and the average frog leaping algorithm in solving optimization problems.

**Key words:** optimization algorithm; shuffled leaping frog algorithm; differential evolution; archive

## 1 引言

混合蛙跳算法(SFLA)<sup>[1]</sup>是由 Eusuff 等人于 2003 年提出的一种模拟青蛙觅食过程的新的智能优化算法. 该算法融合了模因演算算法(memetic algorithm, MA)和粒子群优化算法(particle swarm optimization, PSO)两者的优点, 具有模型简单、易于实现、控制参数少等优点, 近年来已被成功应用于智能优化领域<sup>[2-5]</sup>. 但相关实验测试表明, SFLA 虽具有局部精确搜索的特

点, 却因在寻优过程只利用了全局最优和子群最优青蛙对子群最差青蛙更新使得算法前期容易陷入局部最优, 导致种群多样性降低, 求解精度低, 后期收敛速度慢. 为了提高 SFLA 解决优化问题的性能, 国内外学者对其进行了大量的研究. 如: Elbeltagi 等人<sup>[6]</sup>将“认知分量”引入子群内部搜索策略中, 提高了算法的求解成功率, 一定程度上提高了算法的全局搜索能力; 赵芳等人<sup>[7]</sup>根据适应值所在范围定义新的粒子分类标

<sup>①</sup> 基金项目: 广东省科技计划(2013B051000075)

收稿时间: 2016-04-22; 收到修改稿时间: 2016-06-12 [doi:10.15888/j.cnki.csa.005563]

准避免了算法的盲目搜索,通过动态调整惯性权重提高全局搜索能力,并借用柯西变异算子跳出局部最优的陷阱,从而提高了算法的优化性能;张强等人<sup>[8]</sup>通过动态改变多样性比例来改变子群最优值的多样性密度来增加种群多样性。

本文在借鉴前人研究成果的基础之上,针对SFLA<sup>[9]</sup>易陷入局部最优和收敛速度慢的缺点,根据差分进化算法中的变异、交叉操作不仅能充分利用种群信息从而提高算法的多样性,同时还可以有效地提高算法的搜索速度的特点,提出了一种新的更新策略,在算法前期加入改进的差分算子 rand-1 来更新个体,增加随机扰动性,提高全局搜索能力;在算法后期,加入差分算子 best-rand-2 来提高算法的收敛速度。同时,处理越界个体时对变化尺度进行动态调整,改进了算法的寻优精度。而在每一代更新完成后,引入归档集,保存了好的被替代个体,而被替代的这些个体可能包含有用的信息,有助于收敛到最优点,从而保持了种群的多样性。仿真实验测试结果表明,改进后的差分蛙跳算法(记为DSFLA)较基本SFLA和基于平均值改进算法(记为SFLA-AV)而言,新算法加快了收敛速度,大大提高了求解精度,说明了算法的有效性和可行性。

## 2 混合蛙跳算法

在已知定义空间中随机产生 $N$ 个点组成初始群体 $X=\{X_1, X_2, \dots, X_N\}$ ,第 $i$ 点的位置代表函数在可行域的一个解 $X_i=(x_i^1, x_i^2, \dots, x_i^d)$ ,其中 $d=1, 2, \dots, r$ , $r$ 为解空间的维数。根据目标函数计算出所有青蛙的初始适应值并升序排序,第一只青蛙记为种群的最优青蛙 $X_g=(x_g^1, x_g^2, \dots, x_g^d)$ 。然后,把种群平均分为 $s$ 个子群,每个子群有 $p$ 只青蛙, $N=p \times s$ ,划分原则为

$$S_i = \{S_{ij} | S_{ij} = X_{(i-1)p+j}\}$$

式中, $i=1, 2, \dots, s$ , $j=1, 2, \dots, p$ 。每个子群分别用 $X_b=(x_b^1, x_b^2, \dots, x_b^d)$ 、 $X_w=(x_w^1, x_w^2, \dots, x_w^d)$ 来表示适应值最好的青蛙和适应值最差的青蛙。在子群的每一次进化中,对最差的青蛙 $X_w$ 的位置进行调整,其更新策略为:

青蛙移动的步长:

$$D_j = \text{rand}(1) * (X_b - X_w) \quad (1)$$

青蛙更新后的位置:

$$X_w' = X_w + D_j \quad (-D_{\max} \leq D_j \leq D_{\max}) \quad (2)$$

其中,rand(1)是0~1之间的随机数, $D_{\max}$ 是青蛙允许

移动的最大距离。

对于子群最差青蛙 $X_w$ 位置更新过程中,如果更新策略产生一个较好的解,则用新解 $X_w'$ 更新 $X_w$ ;否则用种群最优的青蛙 $X_g$ 替换子群最优青蛙 $X_b$ 执行公式(1)(2);若仍没有改进,则从定义空间中随机产生一个新解取代 $X_w$ ,这样就完成了子群的一次进化。所有子群按照这种更新策略更新最差个体,直到子群迭代次数。然后各子群的所有个体重新混合,计算适应值按升序排序后重新分组,继续进行局部搜索更新,如此反复直到达到全局最大迭代次数或者满足约束条件,算法停止。

## 3 改进的差分混合蛙跳算法

### 3.1 引进差分算子更新策略

差分进化算法(differential evolution, DE)<sup>[10]</sup>是一类基于群体智能的启发式随机搜索算法。DE类似于遗传算法,存在变异、交叉和选择等多种进化模式<sup>[11]</sup>,为提高种群的多样性和算法的收敛速度,本文在算法进化前期和后期分别借鉴了rand-1和best-rand-2两种模式并进行了改进,使得该算法比基本SFLA具有更好的寻优性能。

在算法前期,为保持种群的多样性,提高全局的搜索能力,不是对子群中的最差个体更新,而是随机选取三个个体,其中一个个体作为目标个体,其他两个个体用来更新移动步长,借鉴差分变异操作<sup>[12]</sup>的思想,引入改进的差分算子rand-1,新的更新策略为:

$$X_w = X_r + F * (X_{r_2} - X_{r_3}), \quad (3)$$

式子中, $X_r$ , $X_{r_2}$ , $X_{r_3}$ 为子群中随机选取的三个个体, $X_w$ 为产生的新个体, $F \in (0, 1)$ 。

在算法后期,为加快算法的收敛速度,有助于收敛到最优点,用子群中最好的个体作为目标个体,随机选取两个个体更新移动步长,引入差分算子best-rand-2,新的更新策略为:

$$X_w = X_b + F * (X_{r_2} - X_{r_3}) \quad (4)$$

式子中, $X_{r_2}$ , $X_{r_3}$ 为子群中随机选取的两个个体, $X_w$ 为产生的新个体, $X_b$ 为子群最优个体 $F \in (0, 1)$ 。

引入改进的差分变异操作后,为了进一步提高算法的局部搜索能力,继续保持种群的多样性,对产生的新个体 $X_w$ 继续执行交叉操作<sup>[12]</sup>,改进后的交叉更新策略如下:

$$x_w^j = \begin{cases} x_w^j, \text{rand}_i^j \geq P_c \text{ or } j = j_{\text{rand}} \\ \text{rand}(0,1), \text{ otherwise} \end{cases} \quad (5)$$

式子中,  $x_w^j$  为当前个体第  $j$  维的值, 其中,  $w=1,2,\dots,N; j=1,2,\dots,d$ . 同时, 每次都对新产生的个体进行改进的越界处理, 处理策略如下:

$$x_w^j = \begin{cases} Lb^j + [1-\text{rand}(1)]^{(1-\frac{CurT}{MaxT})} * [Ub^j - Lb^j], x_w^j < Lb^j \\ Ub^j - [1-\text{rand}(1)]^{(1-\frac{CurT}{MaxT})} * [Ub^j - Lb^j], x_w^j > Ub^j \end{cases} \quad (6)$$

式子中,  $x_w^j$  为当前个体第  $j$  维的值, 其中,  $w=1,2,\dots,N; j=1,2,\dots,d$ ;  $Ub^j$ 、 $Lb^j$  分别指定空间第  $j$  维的上下界;  $CurT$  指当前种群进化代数,  $MaxT$  指种群进化最大代数.

### 3.2 归档集

在子群进化的过程中, 有些被更新掉的个体可能包含有用的信息, 有助于算法收敛到最优点, 因此在子群进化中加入归档集<sup>[13]</sup>可以保存被更新的最差个体. 归档集的具体操作: 在初始化的时候, 随机产生  $2N$  个个体, 建立归档集. 在每一个种群进化中, 每个子群更新次数内淘汰的个体中的一半随机取代归档集里相同数目的个体. 归档集的使用进一步保持了种群的多样性, 提高算法的全局搜索能力.

### 3.3 算法流程

改进的差分混合蛙跳算法具体的流程如下所示.

第一步: 设置相关参数种群规模  $N=200$ , 解空间维数  $VectorSize=5$ , 子群数  $S=20$ , 子群内更新次数  $TeamIter=10$ , 种群最大迭代次数为  $MaxIter=100$ ,  $F=0.4$ ,  $P=0.5$ ;

第二步: 随机初始化种群的每只青蛙, 并根据目标函数计算每只青蛙的适应值;

第三步: 根据青蛙适应值对种群升序排序, 记录第一只青蛙为种群的最优青蛙  $Fbest$ ;

第四步: 将种群按指定规则划分为 20 个子群, 每个子群 10 只青蛙, 并记录每个子群中最优的青蛙和最差的青蛙;

第五步: 建立归档集, 随机产生两倍种群数量的青蛙, 并计算每只青蛙的适应值;

第六步: 按以下规则对每个子群独立进化 10 次, 每一次子群进化完成后产生 20 个被淘汰的个体, 按适应值降序排序, 将前 10 个随机取代归档集中的 10 个个体.

在种群迭代 30%代以前, 根据差分算子  $\text{rand-1}$  按

式子(3)更新得到新个体, 对新个体按式子(5)越界处理, 如果新个体的适应值优于子群最差个体则替代之并按式子(4)进行交叉操作, 再次越界处理; 否则从归档集随机选取三个个体按式子(3)进行更新得到新个体, 越界处理计算适应值, 如果新个体的适应值优于子群最差个体则用新个体替代子群最差个体, 否则从归档集中随机选一个替代之;

在种群迭代 30%代以后, 根据差分算子  $\text{best-rand-2}$  按式子(4)更新并按式子(6)越界处理得到新个体, 如果新个体的适应值优于子群最差个体则替代之并按式子(5)进行交叉操作并越界处理; 否则从归档集随机选取两个个体和子群最优个体按式子(4)更新产生新个体并越界处理计算适应值, 如果新个体的适应值优于子群最差个体则用新个体替代子群最差个体, 否则从归档集中随机选一个替代之;

第七步: 是否达到子群最大迭代次数, 是则完成一次种群迭代, 并将进行第八步, 否则返回第六步;

第八步: 混合子群中所有的青蛙, 重新形成一个完整的种群, 并按适应值升序排序, 记录第一只青蛙为种群的最优青蛙  $Fbest$ , 完成种群的一次更新, 判断是否满足终止条件, 是则输出最优青蛙  $Fbest$  的相关信息, 算法结束; 否则进行种群下一代更新, 跳转第四步.

## 4 实验结果及分析

### 4.1 测试函数与条件

为了验证 DSFLA 的优化性能, 本文选取五个典型的连续优化函数进行测试, 并与基本混合蛙跳算法 (SFLA) 和基于平均值改进算法 (记为 SFLA-AV) 作比较.

为了验证 DSFLA 的优化性能, 本文选取五个典型的连续优化函数进行测试, 并与基本混合蛙跳算法 (SFLA) 和基于平均值改进算法<sup>[14]</sup> (记为 SFLA-AV) 作比较.

$$f_1: f(x) = \sum_{j=1}^5 x_j^2, x_j \in [-1, 1];$$

$$f_2: f(x) = \sum_{i=1}^5 [x_i^2 - 10 \cos(2\pi x_i) + 10], x_i \in [-1, 1];$$

$$f_3: f(x) = \sum_{i=1}^4 (x_i^2 + x_{i+1}^2)^{0.25} [\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1], x_i \in [-4, 4];$$

$$f_4: f(x) = \frac{1}{4000} \sum_{i=1}^5 x_i^2 - \prod_{i=1}^5 \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, x_i \in [-4, 4];$$

$$f_5: f(x) = -20 \exp[-0.2 \sqrt{\frac{1}{5} \sum_{i=1}^5 x_i^2}] - \exp\left[\frac{1}{5} \sum_{i=1}^5 \cos(2\pi x_i)\right] + 20 + e, x_i \in [-30, 30];$$

其中  $f_1$ :Sphere Model 是单峰函数、 $f_2$ :Rastrigin Function、 $f_3$ :Schaffer1 Function、 $f_4$ :Griewand Function 和  $f_5$ :Ackley Function 都是复杂的多峰函数, 它们的理论最优值均为 0. 算法的参数设置为: 种群规模  $N=200$ , 解空间维数  $VectorSize=5$ , 子群数  $S=20$ , 子群内更新次数  $TeamIter=10$ , 种群最大迭代次数为  $MaxIter=100$ ,  $F=0.4$ ,  $P=0.5$ . 为减小偶然性对算法测试结果产生的影响, 每个算法均独立运行 30 次后取平均值, 仿真结果如表 1 所示.

表 1 计算结果

函数	算法	平均最优解	最优解	目标精度	成功率 (%)
f1	SFLA	4.65945e-05	5.3853e-09	$1 \times 10^{-53}$	0
	SFLA-AV	2.70181e-08	1.5790e-26		0
	DSFLA	4.34131e-56	2.6591e-63		100
f2	SFLA	1.29936	0.0018	$1 \times 10^{-36}$	0
	SFLA-AV	2.97235e-03	0		3
	DSFLA	0	0		97
f3	SFLA	7.77887e-01	0.1889	$1 \times 10^{-10}$	0
	SFLA-AV	2.79900e-01	0.0207		0
	DSFLA	7.77515e-13	3.3829e-15		100
f4	SFLA	1.1815e-03	2.0214e-07	$1 \times 10^{-18}$	0
	SFLA-AV	2.4235e-05	0		3
	DSFLA	0	0		100
f5	SFLA	3.5555e-01	0.0074	$1 \times 10^{-14}$	0
	SFLA-AV	6.3219e-02	2.6550e-10		0
	DSFLA	2.4277e-15	8.8818e-16		100

### 4.2 实验结果和分析

从表 1 的求解结果对比看出: DSFLA 的最优解、平均值和求解精度及成功率都明显优于基本的 SFLA 和 SFLA-AV, 说明 DSFLA 算法后期能进行更加精确的局部搜索, 具有更好的稳定性. 就平均最优解求解的精度来说, 在  $f_1, f_2, f_3, f_4, f_5$  函数中, DSFLA 比 SFLA 分别提高了  $1 \times 10^{51}$  倍,  $1 \times 10^9$  倍,  $1 \times 10^{12}$  倍,  $1 \times 10^9$  倍,  $1 \times 10^{14}$  倍; DSFLA 比 SFLA-AV 分别提高了  $1 \times 10^{48}$  倍,  $1 \times 10^9$  倍,  $1 \times 10^{12}$  倍,  $1 \times 10^9$  倍,  $1 \times 10^{13}$  倍, 说明 DSFLA 的求解精度得到有效的提高. 其中, 对于函数  $f_2, f_4$ , DSFLA 均搜索出理论最优解.

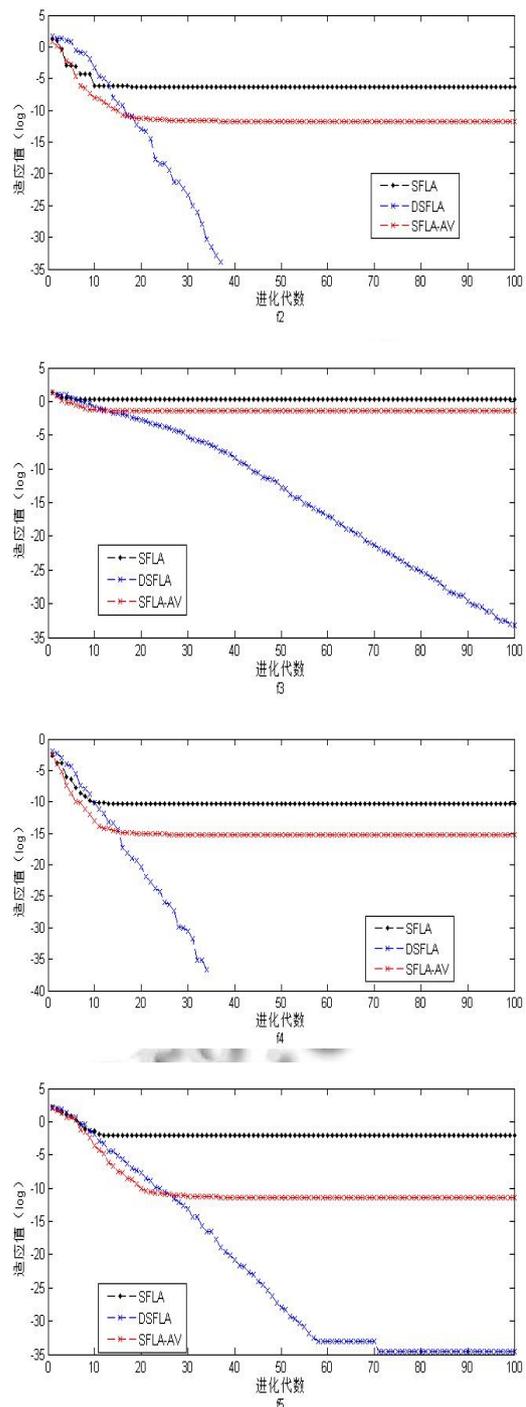
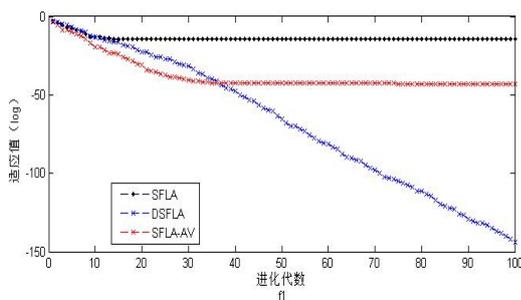


图 1  $f_1 \sim f_5$  的进化曲线

图 1~5 为 3 种算法分别对 5 个典型的连续优化函数搜索最优解的进化曲线. 从图中可以得到: SFLA 和 SFLA-AV 在算法早期就陷入局部最优的陷阱, 后期的收敛速度很慢, 几乎跳不出局部最优的陷阱, 而 DSFLA 在算法进化前期能很好的保持种群的多样性,

提高全部的搜索能力,在算法进化的后期,DSFLA的收敛速度加快,具有能寻得高质量的最优解的能力。

从表格数据和图像的进化曲线都能表明 DSFLA 无论是在求最优解的稳定性上还是质量上都能明显胜于 SFLA 和 SFLA-AV,证明了本文改进的算法是有效和可行的优化算法。

## 5 总结

SFLA 是一种新的智能寻优算法。本文借鉴差分变异的思想,利用子群个体间的信息共享,改进子群最差个体的更新策略,不仅有效的提高了算法的全局寻优能力和求解精度,还加快了算法的收敛速度。算法还通过加入归档集及动态调整越界个体的变化尺度来进一步保持算法的多样性,提高了优化性能。实验仿真结果表明 DSFLA 的有效性和稳定性。

### 参考文献

- 1 Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 2003, 129(3): 210–225.
- 2 郭业才,张苗青.基于混合蛙跳算法的多模盲均衡算法. *兵工学报*,2015,36(7):1280–1287.
- 3 王茜,张粒子,舒隽,王楠.基于阈值选择策略的改进混合蛙跳算法在电网规划中的应用. *电力系统保护与控制*,2011, 39(3):35–39.
- 4 刘紫燕,唐思腾,冯丽,帅暘.混合蛙跳在 AF 协作通信功率优化中的应用. *计算机仿真*,2015,32(7):190–310.
- 5 陈海涛,沈强.改进的蛙跳算法在云计算资源中的研究. *计算机与数字工程*,2015,(8):1382–1506.
- 6 Elbeltagi E, Hegazy T, Grierson D. A modified shuffled frog-leaping optimization algorithm applications to project management. *Structure and Infrastructure Engineering*, 2007, 3(1): 53–60.
- 7 赵芳,张桂珠.基于新搜索策略的混合蛙跳算法. *计算机应用与软件*,2015,(8):224–228.
- 8 张强,刘丽杰,郭昊.一种保持种群多样性的改进混洗蛙跳算法. *计算机与数字工程*,2015,(7):1175–1211.
- 9 Liong SY, Atiquzzaman M. Optimal design of water distribution network using shuffled complex evolution. *The Institution of Engineers*, 2004, 44(1): 93–107.
- 10 Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition based differential evolution. *IEEE Trans. on Evolutionary Computation*, 2008, 12(1): 64–79.
- 11 贺毅朝,王熙熙,刘坤起,王彦祺.差分演化的收敛性分析与算法改进. *软件学报*,2010,21(5):875–885.
- 12 熊伟丽,陈敏芳,王肖,徐保国.运用改进差分进化算法辨识 Hammerstein 模型. *南京理工大学学报*,2013,37(4):536–542.
- 13 王丽,刘玉树,徐远清.基于在线归档技术的多目标粒子群算法. *北京理工大学学报*,2006,26(10):883–887.
- 14 赵鹏军,刘三阳.求解复杂函数优化问题的混合蛙跳算法. *计算机应用研究*,2009,26(7):2435–2437.